UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/769,488	01/29/2004	Shougo Imada	51879/DBP/A400	8744

23363 7590 10/22/2007
CHRISTIE, PARKER & HALE, LLP
PO BOX 7068
PASADENA, CA 91109-7068

EXAMINER	
CRAIG, DWIN M	
ART UNIT	**PAPER NUMBER**
2123	
MAIL DATE	**DELIVERY MODE**
10/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/769,488	IMADA ET AL.	
	Examiner	Art Unit	
	Dwin M. Craig	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 August 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☒ Claim(s) *16* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-24 have been presented for reconsideration based on Applicants' amended claim language and arguments. Claim 25 has been presented for Examination.

Response to Arguments

2. Applicants' arguments presented in the 8/10/07 responses have been fully considered; the Examiner's response is as follows:

2.1 Regarding Applicants' response to the objections to claims 1-24, the Examiner thanks the Applicants' for amending the claims and withdraws the earlier objections to the same.

2.2 Regarding Applicants' response to the 35 U.S.C. 112 second rejections of claim 9, the Examiner thanks the Applicants' for amending claim 1 to provide antecedent support and withdraws the earlier rejection to claim 9.

2.3 Regarding Applicants' response to the 35 U.S.C. 102(b) rejections of claims 1 & 2, on page 11 of the 8/10/2007 responses Applicants' argued;

Vasilko discloses an experimental board, referred to as the RIFLE-62 board, having dynamically reconfigurable FPGAs. Vasilko, however, fails to teach or suggest the use of the RIFLE-62 board in a microcomputer "which replaces a built-in microcomputer incorporated in an existing electronic control unit."

The Examiner respectfully traverses Applicants' argument, Vasilko clearly teaches the use of a microcomputer, see Figure 2 on page 4th page wherein there is disclosed the a "host PC or workstation".

Applicants' further argued on page 11 that;

In addition, Vasilko describes that the RIFLE-62 board includes three types of FPGAs (XC6200, XC3000, and XC4013E), memory, address busses, dual clocks, and dedicated interfaces. The RIFLE-62 board, however, cannot be the claimed "core board" because it fails to include "one or more devices," "computing block," and "second communication block," as is now claimed in amended claim 1.

The Examiner respectfully traverses Applicants' arguments, see Figure 2, "RIFLE-62 board configurations, not the "custom co-processor" variation of the RIFLE-62 board and note the "XC6200 Ext i/f" which is functionally the same as a "second communication block".

Further on page 11 of the 8/10/07 responses Applicants' further argued;

Furthermore, Vasilko fails to teach or suggest a "mother board" coupled to a "core board" over a "peripheral component interconnect (PCI) bus." Although Vasilko discloses a PCI interface, such interface is used to connect the board to a host computer system, and not to the claimed "mother board" where both the "mother board" and "core board" are included in a "microcomputer." Accordingly, claim 1 is now in condition for allowance.

The Examiner respectfully traverses Applicants' arguments, in Figure 2 Vasilko discloses a PCI interface between the Host PC or "motherboard" and the "Core Board" see Figure 1 item "(a) custom co-processor".

On page 12 of the 8/10/2007 responses Applicants' argued:

Vasilko fails to teach or suggest the additional limitations of claim 2. Vasilko discloses a 32-bit PCI interface and a microprocessor interface on the RIFLE-62 board. (See, section 3.5). However, the disclosed interfaces are used to directly write and read data to and from the FPGAs. Nothing in Vasilko teaches or suggests the claimed "bus controller" for controlling the

data flow of data transmitted to and from the core board and mother board via the PCI bus. Accordingly, claim 2 is also in condition for allowance for these added limitations.

The Examiner respectfully traverses Applicants' arguments, Figure 2 in the section labeled, "(a) custom co-processor" teaches a host PC or workstation, this workstation shows a PCI interface slot, and therefore there is a PCI bus controller on the *host PC or workstation.*

In view of Applicants' newly amended claim limitations, the previously applied prior art rejections under 35 U.S.C. 102(b) are being withdrawn.

Claim Objections

3. Claim16 is objected to because of the following informalities: In line 5 of claim 16 reads, *"one or more devices over said bus",* all of the other claims teach a *PCI bus.* Appropriate correction is required.

Claim Rejections - 35 USC § 102/103

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 1, 2 and 25 are rejected under 35 U.S.C. 102(b) as anticipated by or, in the alternative, under 35 U.S.C. 103(a) as obvious over "RIFLE-62: a flexible environment for prototyping dynamically reconfigurable systems" by Milan Vasilko and David Long hereafter referred to as Vasilko.

4.1 Regarding claim 1, Vasilko discloses, *a logic development system using an external microcontroller which replaces a built-in microcomputer incorporated in an existing electronic control unit* (Figure 1 and the descriptive text more specifically, in the section entitled "2.

XC6200-based prototyping environments", "All of the above XC6200 boards were built primarily for reconfigurable computing applications…"),

comprising: a motherboard an application block and a first communication block; (Figure 1 and Figure 2 more specifically example "(d) microprocessor accelerator" and further Figure(s) 3 & 4 the descriptive text in the sections labeled, "5.Software environment" and "6.Prototyping design flow")

a core board including one or more devices which simulate, by software, peripheral devices of the built-in microcomputer so as to execute an input or output process, the core board further including a computing block and a second communication block, a peripheral component interconnect (PCI) bus coupling said mother board and said core board (Figure(s) 3 & 4 the descriptive text in the sections labeled, "5.Software environment" and "6.Prototyping design flow" more specifically the text, "We aimed to provide PCI interface compatibility between RIFLE-62 and XC6200DS[5] to facilitate reuse of available designs…" and Figure 1 under the block labeled XC4013E note the block labeled "32-bit PCI interface"),

and a first communication block, and that is connected to said center block over a bus;(Figure 1 under the block labeled XC4013E note the block labeled "32-bit PCI interface"),

an interface circuit board including circuits associated with hardware of said electronic control unit, wherein the interface board coupled to said electronic control unit, wherein the interface board is coupled to said one or more devices via a harness, wherein; (see the section entitled "3.2 Interconnections" and further the descriptive text which teaches, "Both address and data busses are split into *primary* and *secondary* sections which are connected via fast transceivers, The transceivers also isolate external interface (microprocessor and demo board)

signals from the primary busses..." see the microprocessor interface in the Figure 1 block diagram here is clearly disclose the method of interfacing the "center block" as being the "microprocessor interface" when connected to a microprocessor like an intel I960 as disclosed in the descriptive text, see the section which discloses, "2. XC6200-based prototyping environments ...connected in a mesh, on-board i960 processor...")

said first communication block included in said motherboard and each of said one or more devices included in said core board are coupled to each other over said PCU bus (see Figure 1 "Block diagram of the RIFLE-62 experimental board" see the block labeled XC4013E which discloses a 32-bit PCI interface also see Figure 2 "RIFLE-62 board configurations" see the example "(d) microprocessor accelerator" and "(a) custom co-processor" note the PCI interface),

and said communication block included in said motherboard and each of said one or more devices transfer data to or from each other over said PCI bus, (in Figure 1 note the descriptive text see the section entitled "3.2 Interconnections" and further the descriptive text which teaches, "Both address and data busses are split into *primary* and *secondary* sections which are connected via fast transceivers, The transceivers also isolate external interface (microprocessor and demo board) signals from the primary busses...").

Further and in regards to the claimed "first communication block" Applicants' have argued that this communications block is a PCI bridge/bus controller, *see arguments page 11 of the 8-10-2007 responses*. PCI bridge/bus controllers are well known in the computer architecture art, see U.S. Patent 5,935,223 Figure 1 inside item # 13 "VL-PCI BRIDGE".

4.2 Regarding the rejection of claim 2 see the rejection of independent claim 1 above.

4.3 Regarding claim 25, *Vasilko* teaches or suggests, *wherein the computing block is a microcomputer, and the bus controller in the microcomputer is configured to control flow of information between the mother board and the one or more devices* (see Figure 2 "RIFLE-62 board configurations" and "(a) custom co-processor" note the *host PC or workstation* which is being interpreted to be a *microcomputer* further and in regards to a controller to control flow of data see U.S. Patent 5,935,223 Figure 1 inside item # 13 "VL-PCI BRIDGE" the PCI Bridge will control data flow).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out

the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over RIFLE-62: a flexible environment for prototyping dynamically reconfigurable systems" by Milan Vasilko and David Long hereafter referred to as Vasilko.

5.1 Regarding claim 3, Vasilko does not expressly disclose, wherein: a virtual memory device is interposed between said communication facility included in said motherboard and said PCI bus; and when transfer data is temporarily recorded in said virtual memory device at the timing of receiving or transmitting data, said virtual memory device behaves like a memory device included in an built-in microcomputer.

However, Vasilko does disclose the use of FPGA chips to implement a PCI bus communications system, which must also provide for the various control, data and address registers needed to support a PCI data bus.

Therefore, it would have been obvious to an artisan of ordinary skill, at the time the invention was made to have *Virtual Registers* in the programmed FPGA's as disclosed in Vasilko because otherwise the PCI functionality would not be operable.

6. Claims 4-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over RIFLE-62: a flexible environment for prototyping dynamically reconfigurable systems" by Milan Vasilko and David Long hereafter referred to as Vasilko in view of U.S. Patent 6,356,823 to Iannotti.

6.1 Regarding claim 4, Vasilko does not expressly disclose, wherein: an object on which said application block acts is a vehicle; said logic development system includes an ignition switch; and when said logic development system is interlocked with an on or off state of said ignition switch, control software for said vehicle is initiated or terminated in the same manner as control software residing in said electronic control unit.

However, Iannotti teaches, a core module card (Figure 2 item # 36) being controlled by an ignition switch (Figure 2 item # 86, note the label "ignition switch", see also all of Figure 8a item # 52 and the descriptive text).

Vasilko and Iannotti are analogous art because they are both from the same problem solving area of embedded micro-controller based systems.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to have used the embedded system development teachings of Vasilko with the vehicle monitoring and recording methods of Iannotti.

The motivation for doing so would have been to provide a method of data monitoring and recording as required by current regulatory requirements (see Iannotti Col. 5 lines 25-33).

Therefore, it would have been obvious to combine Vasilko with Iannotti to obtain the invention as specified in claims 4-8.

6.2 Regarding claim 5, Vasilko teaches, wherein: said circuits that are included in said interface board and associated with the hardware of said electronic control unit include at least on facility circuit in which a microcomputer is incorporated;

However, Vasilko does not expressly disclose, said facility circuit is not actuated with the on state of said ignition switch but is actuated synchronously with starting up of the mother board.

However, Iannotti teaches, said facility circuit is not actuated with the on state of said ignition switch but is actuated synchronously with the timing of starting up the center block (Col. 6 lines 41-46 more specifically, *"...include the ability to offer a quick boot-up time, needed to prevent loss of data..."* and Col. 7 lines 40-65).

6.3 Regarding claim 6, Vasilko does not expressly disclose wherein said facility circuit includes a power circuit that is actuated with the on state of said ignition switch, and a logic circuit which actuates the microcomputer in the facility circuit when both a signal sent from said power circuit and a signal sent from said mother board become valid, actuates said microcomputer.

However, Iannotti teaches, (Col. 6 lines 41-46 more specifically, *"...include the ability to offer a quick boot-up time, needed to prevent loss of data..."* and Col. 7 lines 40-65).

6.4 Regarding claim 7, Vasilko does not expressly disclose, wherein: when said ignition switch is turned off, data that should be held is stored in either of a memory included in an external storage device coupled to said logic development system or a memory included in said logic development system; when said ignition switch is turned on, data that should be held is read from said external storage device and restored; and the same capability as the capability of a backup memory is thus realized for said logic development system.

However, Iannotti teaches, Figure 2 item # 74 and more specifically, Col. 7 lines 17-39 and Col. 8 lines 49-58 and Col. 7 lines 55-57.

6.5 Regarding claim 8, Vasilko does not expressly disclose, wherein initial values to ports are set within an initialization routine which is executed on said mother board when said ignition switch is turned on after the power supply of said logic development system is turned on.

However, Iannotti teaches that during initialization the different modules are configured including the communications ports see Figure 16 item # 468 and the descriptive text and more specifically, Col. 3 lines 43-67 and Col. 4 lines 1-23 and Col. 15 lines 1-3 more specifically, "In method path **initiated** by block **370** the user enters parameter information for analog or extra-network device data..." the system in Iannotti discloses the use of an RTOS or (Real Time Operating System) see Col. 6 lines 23-59, it would be obvious to have the *ports* i.e. the communications elements of the system of Iannotti to be initialized as in the initialization of a UART, etc...therefore Iannotti substantially teaches the obvious port initialization as expressly claimed.

7. Claims 9-19 and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over RIFLE-62: a flexible environment for prototyping dynamically reconfigurable systems" by Milan Vasilko and David Long hereafter referred to as Vasilko in view of U.S. Patent 5,864,712 to Carmichael.

7.1 Regarding claim 9, Vasilko does not expressly disclose the specific implementation details of the disclosed operation of the teachings of a PCI controller expressly teaching, *wherein: said PCI bus contains a one-channel interrupt signal line over which an interrupt request is issued from said core board to said mother board; when said interrupt signal line is activated by said core board, said application facility included in said mother board accepts an*

interrupt request; and after the interrupt request is accepted, said interrupt signal line is inactivated.

However, Carmichael teaches, *a one-channel interrupt signal line* (Figure(s) 6A & 6B and Col. 16 line 61 "...its outstanding interrupt flag is set..." and lines 62-67 Col. 17 lines 1-10), *a request issued from the peripheral block to the center block* (Col. 15 lines 48-67 and Col. 16 lines 1-14 more specifically "a new I/O request generated by the CPU...") and *after the interrupt signal is accepted, said interrupt signal line is inactivated* (Figure 6a and the descriptive text, when the interrupt flag is cleared the interrupt line is de-asserted).

Vasilko and Carmichael are analogous art because they are from the same problem solving area of PCI data buses.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to use the interrupt operational methods of Carmichael with the PCI bus methods of Vasilko.

The suggestion for doing so would have been that these methods of servicing interrupts in PCI bus environments are well known in the art and the invention disclosed in Vasilko would not be able to function properly without these known in the art methods of servicing interrupts on a PCI bus, see Carmichael Col. 17 lines 4-10.

Therefore, it would have been obvious to combine Carmichael with Vasilko to obtain the invention as specified in claims 9-19 and 22-24.

7.2 Regarding claim 10, Vasilko does not expressly disclose the specific implementation details, *wherein when interrupt handling is terminated, said application facility included in said mother board checks if said interrupt signal line is inactive.*

However, Carmichael teaches Col. 15 lines 48-55 which clearly teaches software running on a processor (a core) where the interrupt is terminated and further that the interrupt signal line is terminated, the same as the flag being cleared, see Figure(s) 6A and 6B.

7.3 Regarding claim 11, Vasilko does not expressly disclose the specific implementation details, *wherein when interrupt handling is terminated, if said interrupt signal line is active, said application facility included in said mother board inactivates said interrupt signal line.*

However, Carmichael teaches the functional equivalent, see Figures 6A and 6B and the descriptive text and Figure 6C specifically items #'s 604, 605 & 607 and the descriptive text more specifically Col. 15 lines 56-67 and Col. 16 lines 1-45 which clearly details the operation of servicing and then disabling an interrupt service request made by a peripheral (PCI device).

7.4 Regarding claim 12, Vasilko does not expressly disclose the specific implementation details *wherein: said computing facility included in said core board includes a facility for temporarily fetching data; when a large amount of data is transferred between said mother board and each of said one or more devices included in said core board, the large amount of data is transferred in a burst mode between said mother board and said computing block, and transferred in a non-burst mode between said computing facility and each of said one or more devices.*

However, Carmichael teaches the transfer of data between the computing facility and the peripheral block (Figure 1 and the descriptive text), as well as transferring the data using burst mode (Col. 6 lines 24-50 more specifically "...during buffer sized bursts of data...") the non-burst mode is referred to as programmed I/O transfer which PCI bus master devices are capable of programmed during the initialization period.

7.5 Regarding claim 13 Vasilko does not expressly disclose the specific implementation details, *wherein after said application facility included in said mother board accepts an interrupt request, said application facility acquires interrupt flags from each of said one or more devices over said PCI bus; after said application facility acquires interrupt flags, said application facility clears the interrupt flags present in each of said one or more devices.*

However, Carmichael teaches the management and clearing or servicing of interrupts and the interrupt flags during the course of operation of a DMA bus master controller using scatter/gather DMA data transfer (Figure(s) 1, 6 and the descriptive text, more specifically Col. 6 lines 24-50 and Col. 15 lines 56-67 and Col. 16 lines 1-45).

7.6 Regarding claim 14, Vasilko does not expressly disclose the specific implementation details, *wherein after said application facility included in said mother board acquires interrupt flags, said application facility executes a process associated with each of acquired interrupt flags.*

However, Carmichael teaches or makes obvious, acquisition of interrupt flags and executing a process when those flags are acquired (Col. 16 lines 15-45 and Figure 6A, 6B, 6C and more specifically, "...element 662 is operable to set a flag indicating that channel 0 has an outstanding interrupt condition...").

7.7 Regarding claim 15, Vasilko does not expressly disclose the specific implementation details *wherein: after said application block included in said mother board accepts an interrupt request, said application facility acquires a plurality of interrupt flags from each of said core board over said PCI bus; said application facility selects one interrupt flag assigned a high priority, and executes a process associated with the interrupt flag; and after the process is*

completed, said application block clears a process completion interrupt flag present in each of said one or more devices.

However, Carmichael teaches or makes obvious, acquisition of interrupt flags and executing a process when those flags are acquired (Col. 16 lines 15-45 and Figure 6A, 6B, 6C and more specifically, "...element 662 is operable to set a flag indicating that channel 0 has an outstanding interrupt condition...") regarding the clearing of the flags see (Col. 16 lines 42-45 more specifically "...processing continues with element 668 to clear the flag indicating an outstanding interrupt is pending on channel 1...").

7.8 Regarding claim 16, Vasilko does not expressly disclose the specific implementation details *wherein after said application facility selects one interrupt flag assigned a high priority, and executes a process associated with the interrupt flag, said application facility re-acquires a plurality of interrupt flags from each of said one or more devices over said "PCI" bus.*

However, Carmichael teaches or makes obvious, the execution of a process *interrupt service routine* see Col. 15 lines 11-26 more specifically, "...complex interrupt handling software for the host processor..." regarding priority see Col. 16 lines 15-45 and Figure(s) 6A, 6B and 6C.

7.9 Regarding claim 17, Vasilko does not expressly disclose the specific implementation details *wherein said interrupt flags are concurrently stored at successive addresses in one of registers included in each of said one or more devices.*

However, Carmichael teaches the functional equivalent of storing flags in registers by instead storing PRD table entries that are used to obtain outstanding interrupt requests, see Figure 4 item #138 and the descriptive text and Figure(s) 6A, 6B, and 6C.

7.10 Regarding claim 18, Vasilko does not expressly disclose the specific implementation details, *wherein: a plurality of core boards is included; interrupt flags representing interrupts caused by each of a plurality of resources that are included in each of said core boards are stored in a register included in each of said core boards; the interrupt flags representing interrupts caused by each of the resources included in the first core board are stored in the register included in the first core board; and an extension interrupt flag indicating whether interrupt flags, representing interrupts caused by each of the resources included in each of the remaining core boards, are present is stored in association with each core board.*

However, Carmichael teaches or makes obvious, the storing of interrupt flags in registers, it is noted by the examiner that modern Central Processing Units contain interrupt flag tables and further contain registers for storing interrupt vectors or pointers to interrupt service routines, these vector table are used when servicing the interrupts as disclosed in Carmichael, see Figure(s) 6A, 6B and 6C and the descriptive text, more specifically see Col. 15 lines 11-26 more specifically, "...complex interrupt handling software for the host processor..." regarding priority see Col. 16 lines 15-45.

7.11 Regarding claim 19, Vasilko does not expressly disclose the specific implementation details *wherein if said extension interrupt flag demonstrates that interrupt flags are stored in the register included in any of the remaining core boards, said application block acquires the interrupt flags from the register in the remaining core board.*

However, Carmichael teaches or makes obvious the clearing of all interrupt flags scatter/gather DMA as disclosed requires that all blocks of data be transferred from the host PCI controller to memory and that an interrupt be periodically sent to signal that a transfer from

memory or to memory has been completed. Once an interrupt has been serviced the system reacts to the next interrupt, see Figure 5B which describes that process, note item # 157 I/O device and I/O controller signal completion of channel 1 PRD table DMA transfers to processor" which then leads to item # 150 in Figure 6A which discloses, "I/O control device fetches next physical region descriptor entry from the channel 0 PRD table" which then continues to item # 668 "Clearing outstanding interrupt channel 1 flag" which means there is a queue of interrupt flags that are being serviced. Applicants' are merely claiming the mechanisms for storing pending interrupts and how a processor coupled to a PCI bus and devices services those interrupts.

7.12 Regarding independent claim 22, *Vasilko* teaches all of the claimed limitations, as rejected above for independent claims 1 & 2 with the exception that it fails to expressly disclose the specific implementation details, *wherein: when an interrupt factor occurs in any of said one or more devices, said application block reads or writes data in or from said one or more devices; and data whose processing speed is requested to be low is read or written all together during communication performed before or after action of said application block.*

However, Carmichael teaches or makes obvious processing an interrupt, which is functionally the same as an *interrupt factor* (Figure 6A, 6B and 6C and the descriptive text more specifically Col. 15 lines 11-26 more specifically, "...complex interrupt handling software for the host processor...") and regarding transferring of data, i.e. reading or writing data, clearly Carmichael teaches, (Figure 4 and the descriptive text) regarding the limitation *; and data whose processing speed is requested to be low is read or written all together during communication performed before or after the action of said application facility* PCI controllers will buffer the

data as in arrives and store the contents into main memory, the processor regardless of speed, will then be able to process the data after the interrupt is produced by the PCI controller, the end result is that if the processor is executing at a lower speed the data will still be available afterward which is the functional equivalent of Applicants' claimed slower processor.

7.13 Regarding claim 23, *Vasilko* teaches all of the claimed limitations, as rejected above for independent claims 1 & 2 with the exception that it fails to expressly disclose the specific implementation details, *said microcomputer logic development method comprising; issuing an interrupt request from said core board to said motherboard over a one-channel interrupt signal line contained in said PCI bus; accepting the interrupt request when said interrupt signal line is activated via said core board; and inactivating said interrupt signal line after the interrupt request is accepted.*

However, Carmichael teaches or makes obvious *said microcomputer logic development method comprising the steps of; issuing an interrupt request from said peripheral block to said center block over a one-channel interrupt signal line contained in said bus; accepting the interrupt request when said interrupt signal line is activated by means of said peripheral block; inactivating said interrupt signal line after the interrupt request is accepted.*

Carmichael teaches the functional equivalent of acquisition of interrupt flags and executing a process when those flags are acquired (Col. 16 lines 15-45 and Figure 6A, 6B, 6C and more specifically, "...element 662 is operable to set a flag indicating that channel 0 has an outstanding interrupt condition...") regarding the clearing of the flags see (Col. 16 lines 42-45 more specifically "...processing continues with element 668 to clear the flag indicating an outstanding interrupt is pending on channel 1...")

7.14 Regarding claim 24 Vasilko does not expressly disclose the specific implementation details further comprising the steps of: *after an interrupt request is accepted, acquiring interrupt flags from each of said one or more devices over said bus; and after the interrupt flags are acquired, clearing the interrupt flags from each of said one or more devices.*

However, Carmichael teaches, the functional equivalent of acquiring interrupt flags (Figure 6B item # 663 "set outstanding interrupt channel 1 flag"), and over a bus (Figure 1 item # 32 and it is noted that Vasilko teaches a PCI bus) and clearing the interrupt flags at the completion of the processing (Figures 6A, 6B and 6C and the descriptive text).

8. Claims 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over RIFLE-62: a flexible environment for prototyping dynamically reconfigurable systems" by Milan Vasilko and David Long hereafter referred to as Vasilko in view of U.S. Patent 5,908,455 to Parvahan.

8.1 Regarding claim 20, Vasilko does not expressly disclose, wherein: a plurality of core boards are included; the first core board alone includes a free-run timer; said first core board includes at least resources that act synchronously with the timer value of said free-run timer; and the remaining core boards include resources independent of said free-run timer.

However, Parvahan teaches, (Figure 9 "32 *-BIT FREE RUNNING TIMER" see also the descriptive text more specifically Col. 4 lines 64-65 and Col. 11 lines 25-54).

Vasilko and Parvarhan are analogous art because they both come from the embedded system art and the use of PCI busses.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine the FPGA embedded system methods of Vasilko with the automotive data collection methods of Parvarhan.

The motivation for doing so would have been to provide for high speed data analysis using high-speed channels, see Col. 4 lines 7-37 and further it should be observed that the use of the methods of Vasilko would greatly decrease the amount of time required to modify or design the system of Parvarhan.

Therefore, it would have been obvious to combine Parvarhan and Vasilko to obtain the invention as specified in claims 20 and 21.

8.2 Regarding claim 21, Vasilko does not expressly disclose, wherein: *the resources that act synchronously with the timer value of said free-run timer include a comparator and a capture unit; and the resources independent of said free-run timer include a pulse-width modulator (PWM), a communication unit, an A/D converter, and ports.*

However, Parvarhan teaches Ports, capturing Port data and A/D/ converters as well as the functional equivalent of a pulse width modulator (see Figures 4, 5 specifically items # 430 a,b,c,d & e and note that these devices are implemented on an FPGA, Figure 6 and further see Figure 7 which again teaches A/D converters and PLX PCI devices and Figure 8 for the same further see Col. 2 lines 20-47 and Col. 8 lines 16-36).

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dwin M. Craig whose telephone number is (571) 272-3710. The examiner can normally be reached on 10:00 - 6:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul L. Rodriguez can be reached on (571) 272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dwin McTaggart Craig
AU 2123

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100